

## Method of rendering graphical objects

### FIELD OF THE INVENTION

5           The present invention relates to a method of and a device for mapping an input image onto an output image.

This invention may be used in digital apparatuses adapted to render graphical objects such as, for example, video decoders, 3D graphic accelerators, video game consoles, personal digital assistants or mobile phones.

10

### BACKGROUND OF THE INVENTION

Texture mapping is a process for mapping an input image onto a surface of a graphical object to enhance the visual realism of a generated output image including said graphical object. Intricate detail at the surface of the graphical object is very difficult to  
15           model using polygons or other geometric primitives, and doing so can greatly increase the computational cost of said object. Texture mapping is a more efficient way to represent fine detail on the surface of the graphical object. In a texture mapping operation, a texture data item of the input image is mapped onto the surface of the graphical object as said object is rendered to create the output image.

20           In conventional digital images, the input and output images are sampled at discrete points, usually on a grid of points with integer coordinates. The input image has its own coordinate space (u,v). Individual elements of the input image are referred to as "texels." Said texels are located at integer coordinates in the input coordinate system (u,v). Similarly, the output image has its own coordinate space (x,y). Individual elements of the output image are  
25           referred to as "pixels." Said pixels are located at integer coordinates in the output coordinate system (x,y).

The process of texture mapping conventionally includes filtering texels from the input image so as to compute an intensity value for a pixel in the output image. Conventionally, the input image is linked to the output image via an inverse affine transform  $T^{-1}$  that can be  
30           expressed by the following equations:

$$\begin{cases} u = u_{\text{ref}} + \frac{\partial u}{\partial x} dx + \frac{\partial u}{\partial y} dy \\ v = v_{\text{ref}} + \frac{\partial v}{\partial x} dx + \frac{\partial v}{\partial y} dy \end{cases} \quad (1)$$

where  $(u,v)$  are the coordinates of an input current point,  $(u_{\text{ref}},v_{\text{ref}})$  are the coordinates of an input reference point,  $\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}$  and  $\frac{\partial v}{\partial y}$  are affine coefficients, and  $dx = x - x_{\text{ref}}$ ,

$dy = y - y_{\text{ref}}$ .

Similarly, the output image is linked to the input image via a direct affine transform T  
5 that can be expressed by the following equations:

$$\begin{cases} x = x_{\text{ref}} + \frac{\partial x}{\partial u} du + \frac{\partial x}{\partial v} dv \\ v = y_{\text{ref}} + \frac{\partial y}{\partial u} du + \frac{\partial y}{\partial v} dv \end{cases} \quad (2)$$

where  $(x,y)$  are the coordinates of an output current point,  $(x_{\text{ref}},y_{\text{ref}})$  are the coordinates of an output reference point,  $\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial x}{\partial v}$  and  $\frac{\partial y}{\partial v}$  are affine coefficients, and  $du = u - u_{\text{ref}}$ ,

$dv = v - v_{\text{ref}}$ .

10 Equations (1) are called the backward mapping equations (i.e. the input coordinates are expressed as a function of the output coordinates) and equations (2) are called the forward mapping equations (i.e. the output coordinates are expressed as a function of the input coordinates).

As explained above, the output image is made of a plurality of triangles defined by the  
15 positions of their vertices in both output and input coordinate systems. Said positions define a unique affine transform between a triangle in the input image and a triangle in the output image. To generate the output image, each output triangle is scan-converted to calculate the intensity value of each pixel of the triangle on the basis of intensity values of texels.

The texels to be filtered for a given pixel are determined thanks to the inverse affine  
20 transform  $T^{-1}$ . The result of the inverse affine transform of the given pixel is a transformed point in the input image. Texels neighboring said transformed point in the input image are then sampled, weighted and summed to compute the intensity value of the given pixel. This process is repeated for all the pixels in the output image.

In order to make the scan-conversion relatively simple with only incremental  
25 computations, an expensive computational setup is required. This setup computes the affine coefficients  $\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}$  and  $\frac{\partial v}{\partial y}$  and output triangle edge slopes  $a_{01}, a_{02}$  and  $a_{12}$ . If the

coordinates of the vertices of the input triangle are  $(u_0,v_0)$ ,  $(u_1,v_1)$  and  $(u_2,v_2)$  and if the coordinates of the vertices of the output triangle are  $(x_0,y_0)$ ,  $(x_1,y_1)$  and  $(x_2,y_2)$ , said coefficients and slopes are determined from the equations (3) and (4), respectively:

$$\begin{cases} \frac{\partial u}{\partial x} = \frac{(u1 - u0)(y2 - y0) - (y1 - y0)(u2 - u0)}{(x1 - x0)(y2 - y0) - (y1 - y0)(x2 - x0)} \\ \frac{\partial u}{\partial y} = \frac{(x1 - x0)(u2 - u0) - (u1 - u0)(x2 - x0)}{(x1 - x0)(y2 - y0) - (y1 - y0)(x2 - x0)} \\ \frac{\partial v}{\partial x} = \frac{(v1 - v0)(y2 - y0) - (y1 - y0)(v2 - v0)}{(x1 - x0)(y2 - y0) - (y1 - y0)(x2 - x0)} \\ \frac{\partial v}{\partial y} = \frac{(x1 - x0)(v2 - v0) - (v1 - v0)(x2 - x0)}{(x1 - x0)(y2 - y0) - (y1 - y0)(x2 - x0)} \end{cases} \quad (3)$$

$$\begin{cases} a_{01} = \frac{x1 - x0}{y1 - y0} \\ a_{02} = \frac{x2 - x0}{y2 - y0} \\ a_{12} = \frac{x2 - x1}{y2 - y1} \end{cases} \quad (4)$$

A hardware system being able to compute these coefficients requires large arithmetic operators. This increases the silicon area needed for the implementation of said operators and thereafter the price of the hardware system.

## SUMMARY OF THE INVENTION

It is an object of the invention to propose a method of and a device for mapping an input image onto an output image, which is less computational expensive than the one of the prior art.

To this end, the mapping method in accordance with the invention is characterized in that it comprises the steps of:

- determining an inverse affine transform for transforming an intermediate rectangle triangle into an input triangle;

- determining a direct affine transform for transforming the intermediate rectangle triangle into an output triangle;

- applying the inverse affine transform to intermediate points of the intermediate rectangle triangle so as to determine intermediate intensity values corresponding to said intermediate points on the basis of input intensity values of texels; and

- applying the direct affine transform to the intermediate points so as to determine output intensity values of pixels on the basis of the intermediate intensity values.

As it will be explained in more detail hereinafter, the introduction of an intermediate space between the input image and the output image makes possible to determine easily the

inverse affine transform and the direct affine transform, contrary to the prior art where the determination of the inverse or direct affine transform is more complex. As a consequence, the mapping method in accordance with the invention allows silicon area to be saved and less power to be consumed.

5           The present invention also relates to a device implementing such a mapping method and to a portable apparatus comprising said device.

Said invention finally relates to a computer program product comprising program instructions for implementing said mapping method.

10           These and other aspects of the invention will be apparent from and will be elucidated with reference to the embodiments described hereinafter.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

The present invention will now be described in more detail, by way of example, with reference to the accompanying drawings, wherein:

- 15   -       Figure 1 shows an image comprising triangle meshes;
- Figure 2 illustrates a first embodiment of the mapping method in accordance with the invention;
- Figure 3 shows a filtering step based on finite impulse response FIR filters;
- Figure 4 illustrates another embodiment of the mapping method in accordance with  
20   the invention;
- Figure 5 illustrates a particular case of transformation between an intermediate space and an output image;
- Figure 6 shows another embodiment of the mapping method in accordance with the invention to solve the particular case of Figure 5; and
- 25   -       Figure 7 shows a block diagram of a rendering device in accordance with the invention.

## **DETAILED DESCRIPTION OF THE INVENTION**

30           The present invention relates to a method of mapping texture data, i.e. luminance or chrominance data, of an input image split into input triangles onto an output image also split into corresponding output triangles. Input and output triangle meshes can be made of any possible triangle as shown in Figure 1.

As neither output mesh nor input mesh can be regular for a backward mapping corresponding to equations (1) or a forward mapping corresponding to equations (2),

respectively, an additional scan-conversion space (s,t), hereinafter referred to as intermediate space, is introduced. The current positions (x,y) and (u,v) in the output image and in the input image, respectively, are computed on the basis of a current point in said intermediate space.

5 Figure 2 illustrates a first embodiment of the mapping method in accordance with the invention. The intermediate space comprises an intermediate rectangle triangle T0 having three vertices (s0,t0)=(0,0), (s1,t1)=(b,0) and (s2,t2)=(0,a). An input triangle T1 having vertices (u0,v0), (u1,v1) and (u2,v2) is linked to the intermediate rectangle triangle T0 via an inverse affine transform BT that can be expressed by the backward mapping equations (1).

10 Thanks to the use of the intermediate rectangle triangle, the calculation of the affine

coefficients  $\frac{\partial u}{\partial s}$ ,  $\frac{\partial u}{\partial t}$ ,  $\frac{\partial v}{\partial s}$  and  $\frac{\partial v}{\partial t}$  is simplified as expressed by equations (5):

$$\begin{cases} \frac{\partial u}{\partial s} = \frac{u1 - u0}{b} \\ \frac{\partial u}{\partial t} = \frac{u2 - u0}{a} \\ \frac{\partial v}{\partial s} = \frac{v1 - v0}{b} \\ \frac{\partial v}{\partial t} = \frac{v2 - v0}{a} \end{cases} \quad (5)$$

Similarly, an output triangle T2 having vertices (x0,y0), (x1,y1) and (x2,y2) is linked to the intermediate rectangle triangle T0 via an inverse affine transform FT that can be

15 expressed by the forward mapping equations (2). Thanks to the use of the rectangle triangle, the calculation of the affine coefficients  $\frac{\partial x}{\partial s}$ ,  $\frac{\partial x}{\partial t}$ ,  $\frac{\partial y}{\partial s}$  and  $\frac{\partial y}{\partial t}$  is also simplified as expressed by

equations (6):

$$\begin{cases} \frac{\partial x}{\partial s} = \frac{x1 - x0}{b} \\ \frac{\partial x}{\partial t} = \frac{x2 - x0}{a} \\ \frac{\partial y}{\partial s} = \frac{y1 - y0}{b} \\ \frac{\partial y}{\partial t} = \frac{y2 - y0}{a} \end{cases} \quad (6)$$

As a consequence two sets of affine coefficients are computed:

20 - one for the backward mapping between the intermediate space and the input image,  
and

- one for the forward mapping between the intermediate space and the output image. These sets of coefficients are computed in a relatively computation-free setup, saving silicon area and consuming less power.

5           Once the two sets of coefficients have been computed, the mapping method in accordance with the invention is adapted to determine, for an intermediate point  $p0(n)$  of the intermediate rectangle triangle  $T0$ ,  $n$  being an integer, an input transformed point  $p1(n)$  in the input triangle  $T1$  using the inverse affine transform  $BT$ . The input transformed point  $p1(n)$  belonging to the input triangle is in general not located on a grid of texels with integer  
10 coordinates. A filtered intensity value  $I(n)$  corresponding to said point  $p1(n)$  is then derived according to a step of filtering a set of texels of the input triangles surrounding said input transformed point  $p1(n)$ .

As a first example, the filtering step is based on the use of a bilinear filter adapted to implement a bilinear interpolation. Said bilinear interpolation is illustrated by the zoom in  
15 Figure 2 and is based, for example, on the use of the texture values  $I(t1)$  to  $I(t4)$  of the 4 texels  $t1$  to  $t4$  surrounding the input transformed point  $p1(n)$  and on the distance  $a1$  to  $a4$  of said point  $p1(n)$  with respect to said 4 texels. The filtered intensity value  $I(n)$  is then derived as follows:

$$I(n) = \frac{a1.I(t1) + a2.I(t2) + a3.I(t3) + a4.I(t4)}{a1 + a2 + a3 + a4} \quad (7)$$

20 This bilinear interpolation is particularly simple to implement. It will be apparent to the one skilled in the art that the bilinear interpolation can be computed according to other equivalent principles. For example, the coefficients  $a1$  to  $a4$  can also correspond to the surface of four quadrants delimited by a horizontal and a vertical line passing through the input transformed point  $p1(n)$ .

25 As another example, the filtering step is based on the use of two mono-dimensional finite impulse response FIR filters applied sequentially in a horizontal direction and in a vertical direction. The filter taps are determined on the basis of the impulse response  $IR1$  or  $IR2$  which is centered on the input transformed point  $p1(n)$  according to a given direction, i.e. horizontal or vertical, and of the grid  $GR$  of texels with integer coordinates, as illustrated in  
30 Figure 3 in the case of a four taps filter. This results in four coefficients  $h1$  to  $h4$  and  $v1$  to  $v4$  for each FIR filter. The horizontal FIR filter is applied to the four horizontal segments of the set of 16 texels, resulting in 4 intermediate filtered intensity values. Then, the vertical FIR filter is applied to the 4 intermediate filtered intensity values, resulting in a final filtered

intensity value  $I(n)$ . It will be apparent to a person skilled in the art that the number of filter taps is not limited to 4 and that said number may be increased or decreased. Other filters can also be applied, such a polyphase filter for example, depending on the desired visual quality and/or the available computing power.

5

Once the filtered intensity value  $I(n)$  associated with an intermediate point  $p_0(n)$  of the intermediate rectangle triangle  $T_0$  has been determined, the mapping method in accordance with the invention is adapted to determine, for said intermediate point  $p_0(n)$ , an output transformed point  $p_2(n)$  in the output triangle  $T_2$  to which the filtered intensity value is then associated, using the direct affine transform  $FT$ .

10

The mapping method in accordance with the invention then comprises a step of calculating a contribution to a set of pixels surrounding said output transformed point  $p_2(n)$  on the basis of the filtered intensity value  $I(n)$  and of a location of the output point with respect to said set of pixels.

15

As a first example, the calculating step is based on the distance of the output transformed point  $p_2(n)$  with respect to a set of pixels belonging to a grid of pixels with integer coordinates and surrounding said output transformed point. For example, if  $d_1$  to  $d_4$  are the distances between the output transformed point  $p_2(n)$  and the 4 pixels surrounding said point  $p_2(n)$ , as shown in the zoom in Figure 2, the four contributions  $j_1(n)$  to  $j_4(n)$  are calculated as follows:

20

$$j_i(n) = \frac{D - d_i}{D} \cdot I(n), \quad (8)$$

where  $D = d_1 + d_2 + d_3 + d_4$  and where  $i = 1, 2, 3$  or  $4$

Said calculating step can also include a sub-step of weighting the contribution  $j_i(n)$  by a surface  $S_2$  corresponding to the direct affine transform  $FT$  of the unitary surface  $S_0$  associated with the intermediate point  $p_0(n)$ . The contribution is then computed as follows:

25

$$j_i(n) = S_2 \cdot \frac{D - d_i}{D} I(n) = k_i \cdot I(n) \quad (9)$$

For a given pixel, the contributions can come from different output transformed points. Said contributions are added for said given pixel until the sum of the coefficients  $k_i$  is equal to 1.

30

Figure 4 illustrates another embodiment of the mapping method in accordance with the invention. The mapping method comprises a step of determining a first set of coefficients

which defines the inverse affine transform BT, said inverse transform being capable of transforming an intermediate rectangle triangle into an input triangle. Said mapping method also comprises a step of determining a second set of affine coefficients corresponding to the direct affine transform FT, said direct transform being adapted to transform the intermediate rectangle triangle into an output triangle.

Once the two sets of coefficients have been computed, the mapping method in accordance with said another embodiment is adapted to determine, for an intermediate point  $p_0(n)$  of the intermediate rectangle triangle  $T_0$ , an output transformed point  $p_2(n)$  in the output triangle using the direct affine transform FT. Said transform is also adapted to transform a surface  $S_0(n)$  formed by intermediate points into a surface  $S_2(n)$ . Then a pixel  $p_{ix}(n)$  with integer coordinates in the  $(x,y)$  coordinate system belonging to the surface  $S_2(n)$ , if any, is determined. The output vector  $V_2(p,q)$  corresponding to the translation of the output transformed point  $p_2(n)$  towards the translated pixel  $p_{ix}(n)$  is thus computed. The corresponding intermediate vector  $V_0(p',q')$  in the intermediate space is thus derived as follows:

$$p' = \frac{p}{\|d\vec{p}\|} \text{ and } q' = \frac{q}{\|d\vec{q}\|}.$$

A translation of the intermediate point  $p_0(n)$  towards a translated intermediate point  $p'_0(n)$  according to the intermediate vector  $V_0$  is then performed. The mapping method in accordance with the invention is then adapted to determine, for said translated intermediate point  $p'_0(n)$ , an input transformed point  $p'_1(n)$  in the input triangle using the inverse affine transform BT (see continuous arrow in Figure 4). A filtered intensity value  $I(n)$  corresponding to said point  $p'_1(n)$  is then derived from a step of filtering a set of texels of the input triangle surrounding said input transformed point  $p'_1(n)$ , as described before. Said another embodiment only requires a filtering step compared to the first embodiment and thus saves computing resources.

Alternatively, the translated intermediate point  $p'_0(n)$  is not computed. Instead, the mapping method is adapted to determine, for the intermediate point  $p_0(n)$ , an input transformed point  $p_1(n)$  in the input triangle using the inverse affine transform BT (see continuous arrow in Figure 4). An input vector  $V_1(p'',q'')$  corresponding to the intermediate vector  $V_0(p',q')$  in the intermediate space is thus derived as follows:

$$p'' = \frac{\partial u}{\partial s} p' + \frac{\partial u}{\partial t} q' \text{ and } q'' = \frac{\partial v}{\partial s} p' + \frac{\partial v}{\partial t} q'.$$



A translated input transformed point  $p'1(n)$  is then determined using the input vector  $V1(p'',q'')$  and a filtered intensity value  $I(n)$  corresponding to said point  $p'1(n)$  is then derived from a step of filtering a set of texels of the input triangle surrounding said point  $p'1(n)$ , as described before.

5

The mapping method in accordance with the invention creates an intermediate scan-conversion space and is adapted:

- to perform a backward mapping from the intermediate space to the input image so as to determine intermediate intensity values corresponding to points of said intermediate space on the basis of input intensity values of texels of the input image, and
- to perform a forward mapping from the intermediate space to the output image so as to determine intensity values corresponding to pixels of said output image on the basis of the intermediate intensity values.

Said mapping method can be refined according to different embodiments.

- 15 According to a first embodiment of the invention, the intermediate rectangle triangle is defined by lengths  $a$  and  $b$ , which are the lengths of the edges aligned with the  $s$  and  $t$  axis, i.e. opposite to the hypotenuse. These two lengths  $a$  and  $b$  are preferably chosen as a power of 2 in order to simplify the affine coefficients computing.

- In order to assure no hole in the output image the vectors based on affine coefficients
- 20  $\left(\frac{\partial x}{\partial s}, \frac{\partial x}{\partial t}\right)$  and  $\left(\frac{\partial y}{\partial s}, \frac{\partial y}{\partial t}\right)$  are also preferably chosen as having a length lower than 1. For example, the two lengths  $a$  and  $b$  equal to a power of 2 greater than the length of the corresponding edges  $e1$  and  $e2$  in the output triangle.

- In some particular cases, there can be a great distortion between the intermediate rectangle triangle and the output triangle. Figure 5 illustrates such a bad case for the
- 25 computation power requested for scan-conversion. The area of the intermediate triangle is much larger than the output triangle so that the number of scanned intermediate points is unbalanced compared with the actual number of pixels to be computed. That is why, according to another embodiment of the invention, the output triangle is cut into two sub-triangles. A new vertex is then introduced on the longest edge of the output triangle, as
- 30 shown in Figure 6, for example at the basis of one of the triangle altitudes. In Figure 6, the intermediate points  $(s0,t0)$ ,  $(s1,t1)$ ,  $(s2,t2)$  and  $(s3,t3)$  correspond to the output transformed points  $(x0,y0)$ ,  $(x1,y1)$ ,  $(x2,y2)$  and  $(x3,y3)$ , respectively. Then, the scan-conversion of the intermediate triangle is performed. For an intermediate point, an input transformed point in

the input image is determined. The texels surrounding said input transformed point are subsequently filtered, as described before. The position of the output transformed point in the output triangle is also determined. The intermediate points are then re-sampled using an input-driven filter, as described before, said filter generating the intensity values of pixels positioned onto a grid with integer coordinates in the output image. Mapping a rectangle triangle to another one assures a well-balanced ratio between intermediate points and pixels.

Figure 7 shows a block diagram of a rendering device in accordance with the invention. Said rendering device is based on a hardware coprocessor realization. This coprocessor is assumed to be part of a shared memory system. The dynamic memory access DMA unit interfaces the coprocessor with a main memory (not represented). A controller CTRL controls the internal process scheduling. An input memory IM contains a local copy of the input image. An initialization INIT unit accesses geometric parameters, i.e. the vertices of the different triangles, through the DMA unit. From said geometric parameters, the INIT unit computes the affine coefficients and the edge slopes for the scan-conversion process.

These geometric parameters are then processed by the rendering REND unit, which is in charge of scan-converting the intermediate triangle setting-up during the initialization process by the INIT unit. The result of the scan-conversion process is stored in a local output memory OM.

The rendering REND unit comprises a triangle scan TSC unit, which is adapted to compute, for each pixel  $p_0$  of a line of the intermediate triangle, the corresponding location  $p_1$  within the input image and the corresponding location  $p_2$  within the output image. The coordinates of the input point  $p_1$  are subsequently used to address the input memory IM to get a set of texels with integer coordinates  $(u,v)$  to be filtered by a first filter INF unit, resulting in a filtered intensity value. Said filtered intensity value is provided together with the coordinates of the output point  $p_2$  to the inputs of a second filter OUTF unit, which is input driven. Said second filter unit is adapted to compute contributions to pixels of the output image having integer coordinates  $(x,y)$  and surrounding the output point  $p_2$ . The output memory OM stores accumulated contributions. Once all the triangles have been rendered, the output memory is written back into a system memory, a video controller being adapted to access the output image in said system memory to display it.

The mapping method in accordance with the invention can be implemented by means of items of hardware or software, or both. Said hardware or software items can be

implemented in several manners, such as by means of wired electronic circuits or by means of an integrated circuit that is suitable programmed, respectively. The integrated circuit can be contained in a portable apparatus. The integrated circuit comprises a set of instructions. Thus, said set of instructions contained, for example, in a memory of the portable apparatus  
5 may cause the integrated circuit to carry out the different steps of the mapping method. The set of instructions may be loaded into the memory by reading a data carrier such as, for example, a disk. A service provider can also make the set of instructions available via a communication network such as, for example, the Internet.

10 Any reference sign in the following claims should not be construed as limiting the claim. It will be obvious that the use of the verb "to comprise" and its conjugations do not exclude the presence of any other steps or elements besides those defined in any claim. The word "a" or "an" preceding an element or step does not exclude the presence of a plurality of such elements or steps.

15